UNLIMITED

(2)

RSRE MEMORANDUM No. 4176

ROYAL SIGNALS & RADAR ESTABLISHMENT

MATCHING GROUND IMAGES TO MAP DATA ON MIL-DAP

Authors: B C Merrifield, P Simpson, N Smith

PROCUREMENT EXECUTIVE,
MINISTRY OF DEFENCE,
R S R E MALVERN,
WORCS.

RSRE MEMORANDUM No. 4176

UNLIMITED

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 4176

TITLE: MATCHING GROUND IMAGES TO MAP DATA ON MIL-DAP

AUTHORS: B C Merrifield, P Simpson, N Smith*

DATE: June 1988

SUMMARY

Fast programmable parallel processors such as Mil-DAP now offer the prospect of achieving real time signal processing operations with the convenience of software-specified performance and without special purpose hardware.
The problem of accurately registering plan views of the ground with maps has previously been tackled with dedicated hardware. The same operations have now been programmed for Mil-DAP to allow comparative assessment of a programmable implementation. It is concluded that a straightforward implementation on an existing Mil-DAP provides similar size and performance figures to those of prototype dedicated hardware. Although the existing Mil-DAP is more expensive than the custom hardware, cheaper chip set versions are anticipated and such a solution could provide improved mission agility, lower technical risk and allow low cost in-service hardware and software updates as the technology advances.

* Logica Ltd, Cobham, Surrey.

CONTENTS

1 INTRODUCTION

The ICL DAP (Distributed Array Processor) has been in use for some ten years and has a large number of users at several sites in the UK. The potential of the DAP architecture for Digital Signal Processing was recognised within RSRE and in 1982 a jointly funded research contract was started to develop a ruggedised, stand-alone DAP (Mil-DAP) for signal processing applications (initially medium PRF radar processing). The Mil-DAP architecture is described in detail elsewhere [1] but briefly consists of an array of 32x32 single-bit processors each with direct access to 8 or 16 Kbits of RAM, giving a total array store of 1 or 2 Mbytes, with a clock rate of 6.5 MHz. Each processor is connected to its four nearest neighbours and control is via an MCU (Master Control Unit) which broadcasts instructions to the processor array in a Singe Instruction Multiple Data (SIMD) mode. A fast I/O (FIO) double buffered interface provides I/O at 40 Mbytes/sec between an external device and the array store. A video board is also available which provides video output via the FIO. Program development is via a slow interface to an ICL PERQ minicomputer which provides a UNIX based operating environment. The programming languages provided are DAP-Fortran, a parallel development of ANSI Fortran, and APAL (Array Processor Assembly Language). These are totally compatible with the original DAP and allow the user to take advantage of the large amount of library and applications software which has been built up over the last decade or so. A block schematic of the Mil-DAP is shown in Figure 1. Further development of Mil-DAP has been spun-off by ICL to a separate company (Active Memory Technology) who have re-engineered the machine in 2um CMOS with a faster clock (10MHz) and larger array and code stores. A family of these machines (renamed the DAP 500 series) is now available.

Three Mil-DAP's have been in operation within RSRE (SP2,SP4 and AD2) since early 1986 and have been applied to a large number of problems [2,3] including radar processing, speech processing, terrain modelling, image processing, image understanding, radar ESM, track-plot correlation and learning machines. The particular application described in the present paper is that of map/image correlation, in which an IR or radar image is processed to extract features such as roads, field boundaries etc. and then compared with a stored map. Current solutions have taken the form of custom built processors, general purpose serial processors being too slow for real time operation. However, the advent of Mil-DAP whose architecture lends itself particularly well to image processing operations, coupled with its ability to trade accuracy for speed has offered the possibility of a programmable solution with all the consequent flexibility and reduced technical risk.

The present paper describes an investigation into the suitability of the DAP for map/image correlation. The image processing and correlation algorithms employed are described and their performance assessed.

2 IMAGE PROCESSING

In the present investigation the map (Fig. 2) is a 384x384 binary map and the image (Fig. 3) is a 128x128 8-bit Infra-Red LineScan (IRLS) image of a road junction within the map at the same scale and orientation. The processing performed on the image prior to correlation can typically consist of edge detection, amplitude limitation and smoothing and were suggested as representative of the type of processing performed by existing custom built

hardware. Figure 4 shows the IRLS image after this processing. Because of array store limitations on our (1 Mbyte) DAP we have had to restrict the map size to 256x256. As in the hardware we perform the correlation at half resolution. The majority of the programming is in DAP-Fortran and a considerable improvement in performance could be expected by programming critical sections in APAL.

2.1 Data Mappings

Most practical problems do not map directly onto the DAP, pixel arrays in image processing for example will normally be many times larger than the DAP array. For problems such as these some way of mapping the problem onto the DAP is required. The most obvious and simplest method is the 'sheet' or 'sliced' mapping (Figure 5a) where the image is simply sliced into a number of 32x32 subimages. A disadvantage of this is the boundary problem where neighbouring pixel elements in different sheets may be many planes away in the DAP array store. An alternative is the so-called 'crinkled' mapping (Figure 5b) in which sub-areas of neighbouring pixels are stored 'under' one PE (Processing Element) thereby eliminating boundary difficulties and reducing the number of data shifts required for neighbouring pixel access. The decision as to which mapping is used is largely dictated by the type of problem. For problems where the global structure is more important than the local, the sliced mapping would be used, whereas if local structure is important the crinkled mapping would be used.

2.2 Edge Detection

The first operation performed on the image is edge detection using the sum of the moduli of the 3x3 Sobel operators in the vertical and horizontal directions, i.e

$$\begin{vmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{vmatrix} \quad + \quad \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}$$

This operation could be programmed for a complete 32x32 pixel sheet in a single DAP-Fortran statement, however, boundaries between sheets can then cause addressing problems . The approach adopted here was to eliminate these edge effects by using 'crinkled' mappings and to calculate the [1 2 1] sums by adding nearest neighbours and then adding neighbouring results. Performance figures are given in section 4 although reductions in time of as much as an order of magnitude could probably be achieved by careful coding in APAL. However, as this application is totally dominated by the correlation calculation no attempt was made to optimise the edge detection routine further.

2.3 Amplitude Limitation

In order to remove unwanted noise from the image it is required to replace the pixels of highest and lowest brightness by 'cut-off' values. Initial solutions centered around a cumulative histogram approach where the histogram 'bins' contained the number of pixels at or below a certain level. The bins were then searched until the appropriate number of pixels at the upper and lower levels was ascertained. All pixels whose values were above the upper limit were set to the upper limit value and those pixels below the lower limit were set to the lower limit value. However, as the histogram was only being used to find the cut-off levels, alternative algorithms were considered. One of these, a

successive approximation method, proved to be almost twice as fast. An initial value (the middle value of the possible pixel value range) is used to calculate the number of pixels in the image whose values are above (and below) that value (here we are able to take advantage of the parallelism of the DAP to compare 1024 pixel values simultaneously). These are then compared with the number of pixels corresponding to the prescribed percentiles. Depending on the results of these comparisons revised approximations for the upper and lower threshold value incrementing (or decrementing) by a value equal to one half of the remaining range until no further improvement is possible. The algorithm was coded in DAP-Fortran and again speed-ups could be achieved by APAL coding if required. The algorithm is independent of the data mapping.

2.4 Smoothing

The final stage of processing is usually the removal of localised peaks and troughs with a smoothing convolution. Typical 3x3 and 5x5 convolution matrices used are given below:

$$\begin{bmatrix} 2 & 3 & 2 \\ 3 & 12 & 3 \\ 2 & 3 & 2 \end{bmatrix} \qquad \begin{bmatrix} 3 & 3 & 4 & 3 & 3 \\ 3 & 7 & 9 & 7 & 3 \\ 4 & 9 & 12 & 9 & 4 \\ 3 & 7 & 9 & 7 & 3 \\ 3 & 3 & 4 & 3 & 3 \end{bmatrix}$$

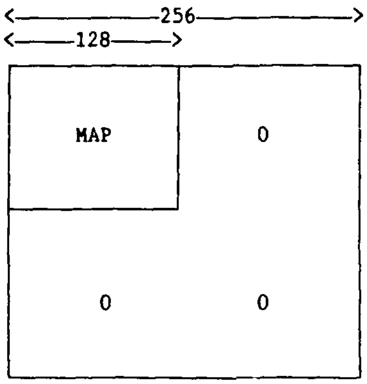
The convolutions were coded in DAP-Fortran using crinkled mapping with APAL subroutines to perform the arithmetic at bit level (to avoid using 16 bits throughout) and using binary shifts instead of multiplication. Figure 4 shows the image after edge detection, amplitude limitation and smoothing with the 3x3 operator. Run times for both 3x3 and 5x5 operators are given in section 4.

2.5 Resolution reduction

In order to reduce the processing time, correlation is initially performed at one half resolution. A full resolution correlation can then be made on a small area of interest. To reduce the resolution by one half, the algorithm adopted was to replace each 2x2 sub-image by a single pixel whose value is the average of the values in the sub-image. The operation was coded in DAP-Fortran with an APAL subroutine to perform the bit-level arithmetic.

3 CORRELATION

Correlating a binary map (mainly zero's) with an image suggests a direct form of correlation algorithm as likely to be efficient since it avoids actual multiplications. However the large number of data shifts necessary to bring together the contributions to each correlation value make the direct approach slower than using two-dimensional Fast Fourier Transforms (FFTs). An FFT subroutine package [4], written in DAP-Fortran using 32-bit floating point arithmetic was used, but because of storage restrictions was modified to use 24-bit floating point arithmetic. The 128x128 half resolution map and the 64x64 half resolution image were stored as 24-bit floating point complex numbers within two 256x256 element arrays, the remainder of the array being padded-out with zero's as shown below (for the map):



The operations performed are the FFTs of the map and image, the complex multiplication of the resulting transforms and an inverse FFT. A 3-dimensional plot of the correlation surface is shown in Figure 6. The displacement of the peak from the origin was calculated and used to position the image on the map. Figure 7 is a composite picture showing the map and image (to the same scale), the 2-dimensional correlation surface (compressed scale and wraparound) and the displaced image superimposed on the map.

4 PERFORMANCE

The following table gives execution times for the individual image processing operations on a 128x128 8-bit image, a 256x256 2D FFT using 24-bit floating point arithmetic and the correlation of a half resolution image (64x64) and map (128x128) also using 24-bit floating point for the two FFTs and the complex multiplication. The time required by custom hardware to perform the map/image correlation task is 4 to 5 seconds. On the Mil-DAP the corresponding time is about 4.5 seconds which is dominated by the correlation (it should be remembered, however, that this is for a 128x128 half-resolution map instead of a 192x192). Off the shelf software has been used for the FFT with no attempt at optimisation beyond the conversion to 24-bits. In addition, it should be noted that the map transformation would in practice be stored, the time for subsequent correlations would then be reduced to approximately 3 seconds.

Process	Code	Run time
Sobel edge detection	Fortran	9.0 ms
Amplitude Limitation	Fortran	14.9 ms
3x3 Smoothing Convolution	APAL	2.9 ms
5x5 Smoothing Convolution	APAL	8.7 ms
Resolution Reduction	APAL	0.17 ms
2D FFT (256x256,24-bit fp)	Fortran	1.49 sec
Correlation	Fortran	4.50 sec

5 CONCLUSIONS

Mil-DAP is a ruggedised, flyable, general purpose signal processor and has been shown capable of performances comparable with that of fixed function custom hardware which has a component cost of around £50K. The size of both prototypes is similar (custom 88 litres and DAP 65 litres including power supplies etc.) and while Mil-DAP was not intended for volume production, a range of fully compatible re-engineered AMT versions are available. The unit cost of a basic AMT DAP 510 (32x32 array) with a 10 MHz clock and 4MBytes of array store (4 times as much as Mil-DAP) is around £95K at the time of writing. The size of the custom prototype could be reduced by VLSI to around 3-5 litres, but similarly, chip set versions of the AMT DAP are likely to be available in the near future. In addition, faster (15 MHz) and larger (64x64,128x128) AMT DAPs are planned.

Very little software optimisation has been carried out on the matching problem and performance improvements can be achieved by using variable precision arithmetic and assembly level programming (factors of 10 improvement in execution speed have commonly been noted in going from DAP-Fortran to APAL). Because DAP is a programmable processor, algorithms can be easily changed or alternatively hardware can be updated without rewriting the software. Since the processing here is likely to occupy only 5sec (less with APAL coding) at intervals, the DAP could be devoted to other mission requirements for most of the time, a strong advantage over dedicated hardware. Alternatively a smaller size of DAP would be adequate and cheaper. As technology develops these arguments will be enhanced.

The versatility of the DAP has been well established [2] in a range of relevant military applications. For example, it has been shown that Mil-DAP can handle a computationally demanding 'medium PRI' airborne radar processing problem with processing capacity in hand, its image processing capabilities suggest possible uses for head-up displays and its performance for connected word recognition in single user, limited vocabulary environments, for cockpit voice recognition.

In view of the advantages of a programmable solution in improving mission agility, reducing technical risk and allowing low cost in-service hardware and/or software updates as the technology advances, a DAP solution must merit serious consideration for implementing real time map/image correlation.

Acknowledgement
The authors are indebted to Mr D.J.Blundell for his help and advice.

References

1. Hunt D.J.,Reddaway S.F. 'Distributed processing power in memory' from 'The fifth generation computer project', G.G.Scarrott (ed), Pergamon Infotech Ltd, Maidenhead 1983.

2. Simpson P.,Roberts J.B.G.,Merrifield B.C., 'Mil-DAP: its architecture and role as a real time airborne digital signal processor' Agard Conf. Proc. No. 380, Lisbon 20-23 May 1985, pp 23-1 to 23-18.

3. Roberts J.B.G.,Harp J.G.,Merrifield B.C.,Simpson P.,Ward J.S.,Webber H.C.,Palmer K.J., 'Highly parallel SIMD and MIMD programmable processors for signal processing and simulation in defence'. IDEX 87, Brighton, Sept. 1987.

4. FFT Software Users Guide. Logica, 1986.

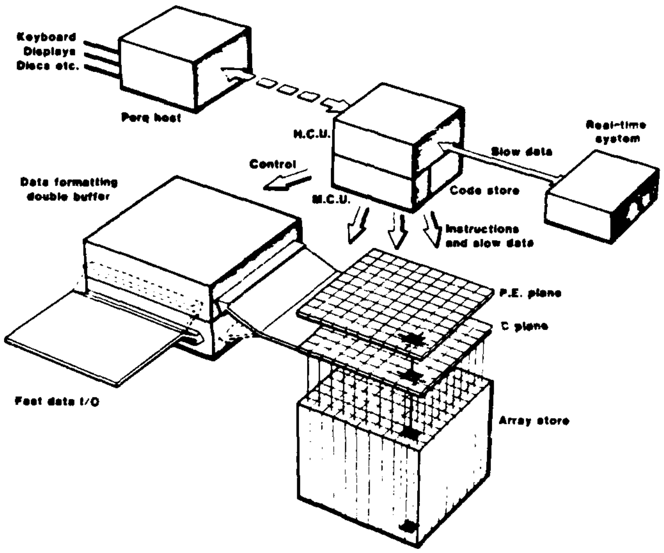


Figure 1. Schematic representation of Mil-DAP showing the relationships between the Host Connection Unit (HCU), Master Control Unit (MCU) and the processing element (PE) plane; and the D input/output plane and the fast data I/O (FIO) unit.

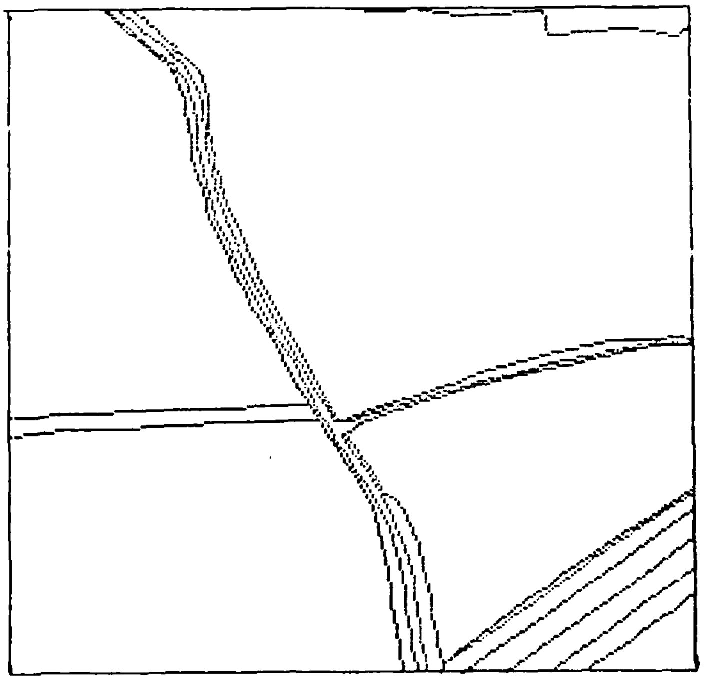


Figure 2. Binary Map (256x256).

Figure 3. IRLS Image before processing (128x128).

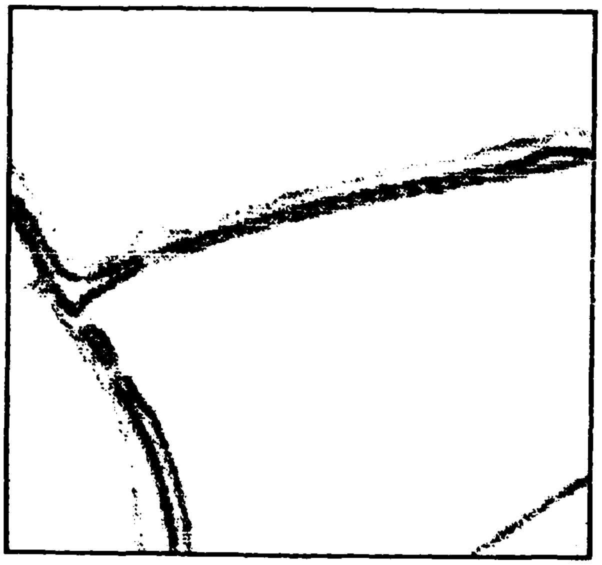
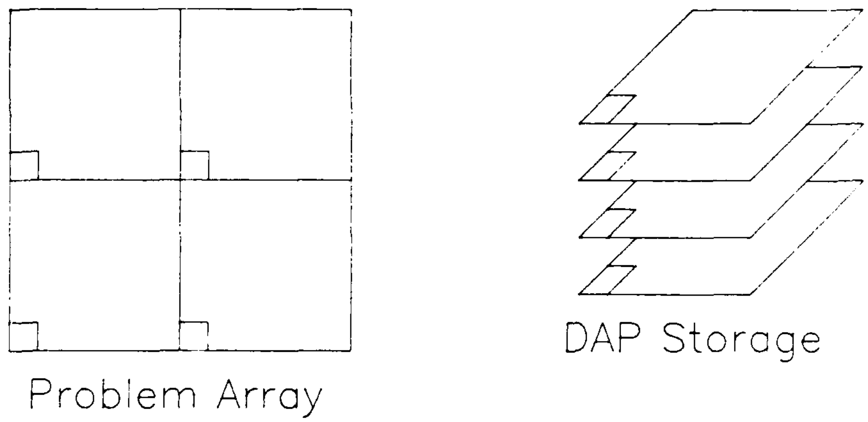
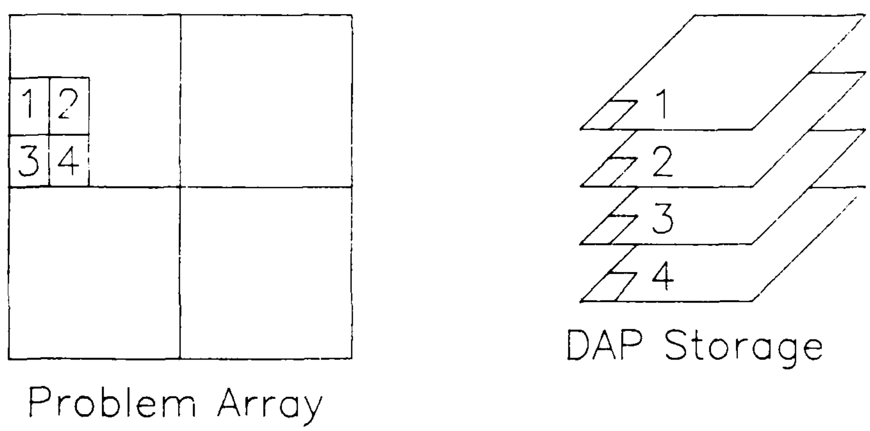


Figure 4. IRLS Image after edge detection,amplitude limitation and smoothing (128x128).



5a) Sliced Mapping



5b) Crinkled Mapping

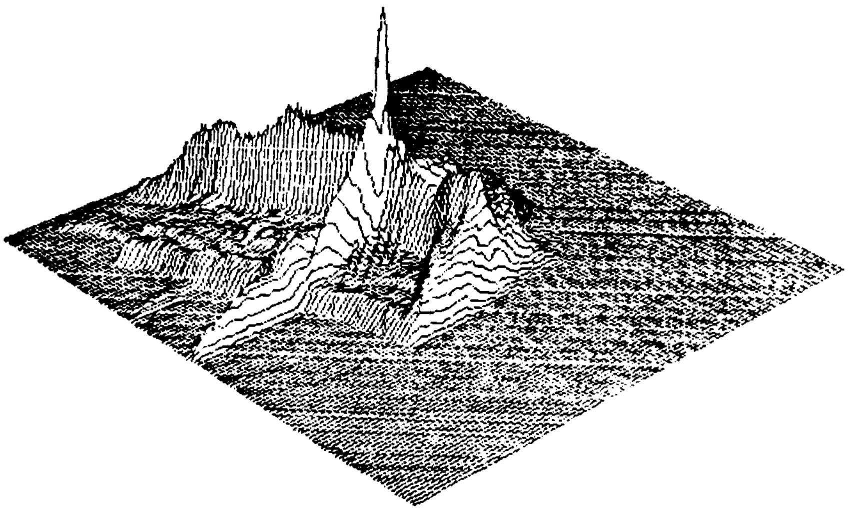


Figure 6. 3D Correlation Surface plot.

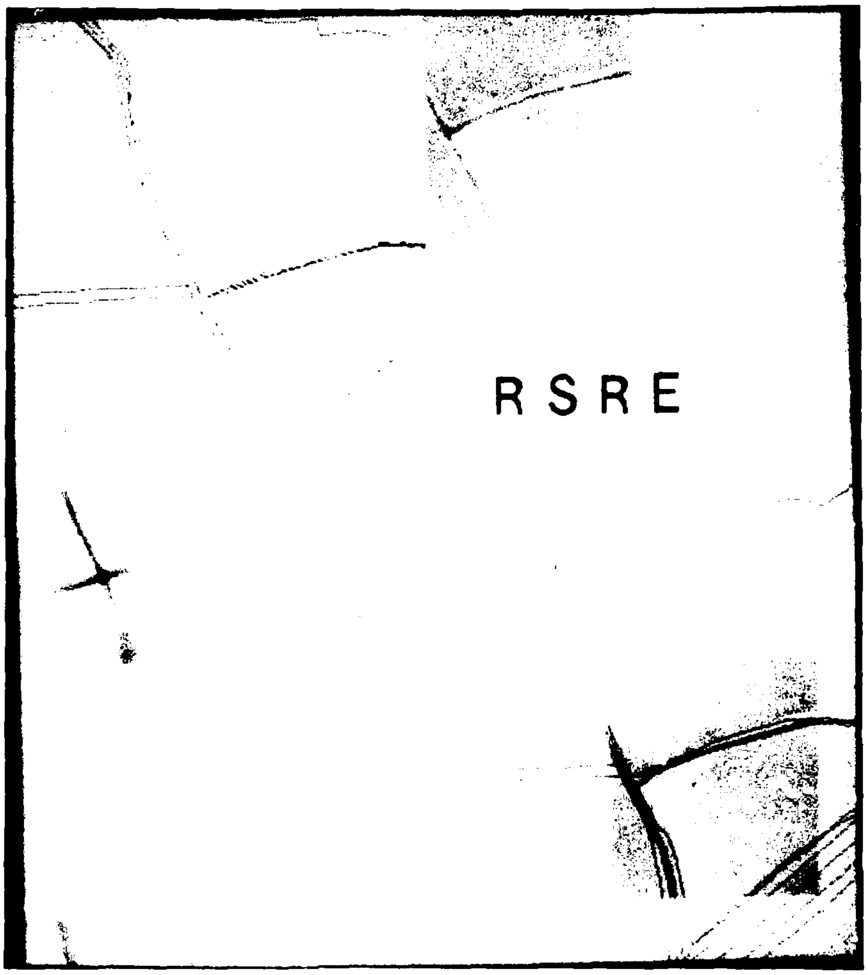


Figure 7. Map, Image, 2D correlation surface and Map with superimposed image.

DOCUMENT CONTROL SHEET

Overall security classification of sheet ..UNCLASSIFIED..

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification
	MEMO 4176		U/C

5. Originator's Code (if known)	6. Originator (Corporate Author) Name and Location
7784000	ROYAL SIGNALS & RADAR ESTABLISHMENT ST ANDREWS ROAD, GREAT MALVERN, WORCESTERSHIRE WR14 3PS
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title Matching ground images to map data on MIL-DAP

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

8. Author 1 Surname, initials	9(a) Author 2	9(b) Authors 3,4...	10. Date	pp. ref.
MERRIFIELD B C	SIMPSON P	SMITH N	1988.6	13

11. Contract Number	12. Period	13. Project	14. Other Reference

15. Distribution statement

UNLIMITED

Descriptors (or keywords)

continue on separate piece of paper

Abstract

Fast programmable parallel processors such as Mil-DAP now offer the prospect of achieving real time signal processing operations with the convenience of software-specified performance and without special purpose hardware.

The problem of accurately registering plan views of the ground with maps has previously been tackled with dedicated hardware. The same operations have now been programmed for Mil-DAP to allow comparative assessment of a programmable implementation. It is concluded that a straightforward implementation on an existing Mil-DAP provides similar size and performance figures to those of prototype dedicated hardware. Although the existing

/cont.../Mil-DAP

S80/48

Mil-DAP is more expensive than the custom hardware, cheaper chip set versions are anticipated and such a solution could provide improved mission agility, lower technical risk and allow low cost in-service hardware and software updates as the technology advances.